

PORTAL

USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

generat* jit IR processor

Searching within **The ACM Digital Library** for: generat* jit IR processor ([start a new search](#))

Found 23 of 264,269

REFINE YOUR SEARCH

▼ Refine by Keywords

generat* jit IR processor

Discovered Terms

▼ Refine by People

Names

Institutions

Authors

Reviewers

▼ Refine by Publications

Publication Year

Publication Names

ACM Publications

All Publications

Content Formats

Publishers

▼ Refine by Conferences

Sponsors

Events

Proceeding Series

ADVANCED SEARCH

Advanced Search

FEEDBACK

Please provide us with feedback

Found 23 of 264,269

Search Results

Related Journals

Related SIGs

Related Conferences

Results 1 - 20 of 23

Sort by

Save results to a Binder

1 Code generation for just-in-time compiled mobile collector agents

John G. Allen, Jesse S. Jin

May 2003 **VIP '02: Selected papers from the 2002 Pan-Sydney workshop on Information and communication technologies**, Volume 22

Publisher: Australian Computer Society, Inc.

Full text available: Pdf (38.50 KB)

Additional Information: [full citation](#), [abstract](#)

Bibliometrics: Downloads (6 Weeks): 2, Downloads (12 Months): 13, Downloaded by 0 users

This paper describes MGGEN/x86, a toolkit that simplifies the process of generating code for just-in-time (JIT) compilers for the x86 series of processor. MGGEN produces portable assembly language code for a user-defined sequence ...

Keywords: JIT, assemble, collector, compiler, mobile, x86

2 Dynamic optimization for efficient strong atomicity

Florian T. Schneider, Vijay Menon, Tatjana Shpeisman, Ali-Reza Adl-Tabatabai, Michael D. Ernst, and Michael L. Scott October 2008 **OOPSLA '08: Proceedings of the 23rd ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**

Publisher: ACM

Full text available: Pdf (342.68 KB)

Additional Information: [full citation](#), [abstract](#)

Bibliometrics: Downloads (6 Weeks): 21, Downloads (12 Months): 196, Downloaded by 0 users

Transactional memory (TM) is a promising concurrency control alternative to locks. However, TM faces important memory model issues regarding TM semantics and exposed pointers. In this paper, we propose a safe, managed languages such ...

Keywords: code generation, compiler optimizations, dynamic optimization, memory management, virtual machines

Also published in:

October 2008 **SIGPLAN Notices** Volume 43 Issue 10

3 Secure virtual architecture: a safe execution environment for commodity servers

John Criswell, Andrew Lenhardt, Dinakar Dhurjati, Vikram Adve, and David R. Andersen October 2007 **SOSP '07: Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles**

Publisher: ACM

Full text available: Pdf (24:37 MIN), Pdf (383.30 KB) Additional Information: [full citation](#), [abstract](#)

Bibliometrics: Downloads (6 Weeks): 31, Downloads (12 Months): 287, Downloaded by 1 user

This paper describes an efficient and robust approach to provide a *safe* system, such as Linux, and all its applications. The approach, which we define ...

Keywords: compiler, memory safety, operating systems, security, type machine

Also published in:

October 2007 **SIGOPS Operating Systems Review** Volume 41 Issue 6

4 Design of the Java HotSpot™ client compiler for Java 6

 Thomas Kotzmann, Christian Wimmer, Hanspeter Mössenböck, Thomas Roser-Möller
May 2008 **Transactions on Architecture and Code Optimization (TACO)**

Publisher: ACM 

Full text available:  Pdf (1.14 MB)

Additional Information: [full citation](#), [abs](#)

Bibliometrics: Downloads (6 Weeks): 40, Downloads (12 Months): 433, Downloaded by 1 user

Version 6 of Sun Microsystems' Java HotSpot™ VM ships with a redesign that includes several research results of the last years. The client compiler used by default ...

Keywords: Java, compiler, deoptimization, intermediate representation, register allocation

5 Compiler and runtime support for efficient software transactional memory

 Ali-Reza Adl-Tabatabai, Brian T. Lewis, Vijay Menon, Brian R. Murphy, Brad Calder
June 2006 **PLDI '06: Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation**

Publisher: ACM 

Full text available:  Pdf (211.55 KB)

Additional Information: [full citation](#), [abs](#)

Bibliometrics: Downloads (6 Weeks): 27, Downloads (12 Months): 373, Downloaded by 1 user

Programmers have traditionally used locks to synchronize concurrent access. While locking is a simple mechanism, synchronization, however, has well-known pitfalls: using locks for fine-grained synchronization that already uses locks are both difficult ...

Keywords: code generation, compiler optimizations, locking, synchronization, machines

Also published in:

June 2006 **SIGPLAN Notices** Volume 41 Issue 6

6 RPython: a step towards reconciling dynamically and statically typed languages

 Davide Ancona, Massimo Ancona, Antonio Cuni, Nicholas D. Matsakis
October 2007 **DLS '07: Proceedings of the 2007 symposium on Dynamic languages**

Publisher: ACM

Full text available: Pdf (239.48 KB)

Additional Information: [full citation](#), [abs](#)

Bibliometrics: Downloads (6 Weeks): 7, Downloads (12 Months): 91, Download

Although the C-based interpreter of Python is reasonably fast, implementation offers some advantages in terms of robustness and interoperability. Unfortunately it is primarily designed to execute ...

Keywords: .NET, JVM, Python

7 C and tcc: a language and compiler for dynamic code generation

Massimiliano Poletto, Wilson C. Hsieh, Dawson R. Engler, M. Frans Kaashoek
March 1999 **Transactions on Programming Languages and Systems**

Publisher: ACM

Full text available: Pdf (471.68 KB)

Additional Information: [full citation](#), [abs](#)

Bibliometrics: Downloads (6 Weeks): 18, Downloads (12 Months): 113, Download

Dynamic code generation allows programmers to use run-time information to generate code that is more expressive and efficient than static code. The 'C(Tick C) language is a high-level ...

Keywords: ANSI C, compilers, dynamic code generation, dynamic code

8 Online optimizations driven by hardware performance monitoring

Florian T. Schneider, Mathias Payer, Thomas R. Gross
June 2007 **PLDI '07: Proceedings of the 2007 ACM SIGPLAN conference on programming language implementation**

Publisher: ACM

Full text available: Pdf (224.36 KB)

Additional Information: [full citation](#), [abs](#)

Bibliometrics: Downloads (6 Weeks): 17, Downloads (12 Months): 122, Download

Hardware performance monitors provide detailed direct feedback about the source of information that a compiler may use for optimization. A JIT compiler can use such information because ...

Keywords: Java, dynamic optimization, hardware performance monitoring

Also published in:

June 2007 **SIGPLAN Notices** Volume 42 Issue 6

9 Optimistic parallelism benefits from data partitioning

Milind Kulkarni, Keshav Pingali, Ganesh Ramanarayanan, Bruce Walter, Kaushik Datta
March 2008 **ASPLOS XIII: Proceedings of the 13th international conference on architectures for parallel languages and operating systems**

Publisher: ACM

Full text available: Pdf (22:0 MIN), Ebook (356.09 KB) Additional Information: [full citation](#), [abs](#)

Bibliometrics: Downloads (6 Weeks): 15, Downloads (12 Months): 223, Download

Recent studies of irregular applications such as finite-element mesh generation have shown that these applications have a generalized data parallelism arising from computations on ...

Keywords: data partitioning, irregular programs, locality, lock coarsening, decomposition

Also published in:

March 2008 **SIGARCH Computer Architecture News** Volume 36 Issue 1

March 2008 **SIGPLAN Notices** Volume 43 Issue 3

March 2008 **SIGOPS Operating Systems Review** Volume 42 Issue 2

10 [Code Generation and Optimization for Transactional Memory Constructors](#)

Cheng Wang, Wei-Yu Chen, Youfeng Wu, Bratin Saha, Ali-Reza Adl-Tabatabai

March 2007 **CGO '07: Proceedings of the International Symposium on Code Generation and Optimization**

Publisher: IEEE Computer Society

Full text available:  [Pdf](#) (365.03 KB) Additional Information: [full citation](#), [abstract](#)

Bibliometrics: Downloads (6 Weeks): 10, Downloads (12 Months): 153, Download Rank: 1000

Transactional memory offers significant advantages for concurrency control. However, the design and implementation of transactional memory constructs in application languages pose a unique set of challenges ...

11 [Just-In-Time compilation on ARM processors](#)

 Michele Tartara, Simone Campanoni, Giovanni Agosta, Stefano Crespi Reghetti

July 2009 **ICOOOLPS '09: Proceedings of the 4th workshop on the Implications of Object-Oriented Languages and Programming Systems**

Publisher: ACM 

Full text available:  [Pdf](#) (604.92 KB) Additional Information: [full citation](#), [abstract](#)

Bibliometrics: Downloads (6 Weeks): 15, Downloads (12 Months): 41, Download Rank: 1000

This paper presents a Just-In-Time compilation system for ARM processors starting from static compilation of the sources into CIL (Common Intermediate Language), an intermediate language that is used ...

Keywords: ARM, dynamic compilation, embedded systems

12 [Tracing for web 3.0: trace compilation for the next generation web application environments](#)

 Mason Chang, Edwin Smith, Rick Reitmaier, Michael Bebenita, Andreas Gal, Franziska Koenig

March 2009 **VEE '09: Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments**

Publisher: ACM 

Full text available:  [Pdf](#) (647.16 KB) Additional Information: [full citation](#), [abstract](#)

Bibliometrics: Downloads (6 Weeks): 53, Downloads (12 Months): 291, Download Rank: 1000

Today's web applications are pushing the limits of modern web browsers. The platform of choice for rich client-side applications has shifted the use of programs to large computationally ...

Keywords: dynamic compilation, dynamically typed languages, forth, j specialization

13 Enforcing isolation and ordering in STM

 Tatiana Shpeisman, Vijay Menon, Ali-Reza Adl-Tabatabai, Steven Balensiefen, Katherine F. Moore, Bratin Saha

June 2007 **PLDI '07: Proceedings of the 2007 ACM SIGPLAN conference**
implementation

Publisher: ACM  [Request Permissions](#)

Full text available:  Pdf (257.39 KB)

Additional Information: [full citation](#), [abs](#)

Bibliometrics: Downloads (6 Weeks): 17, Downloads (12 Months): 208, Downl

Transactional memory provides a new concurrency control mechanism to synchronize. High-performance software transactional memory (STM) atomicity: Accessing shared ...

Keywords: code generation, compiler optimizations, escape analysis, is, transactional memory, virtual machines, weak atomicity

Also published in:

June 2007 **SIGPLAN Notices** Volume 42 Issue 6

14 Hardware atomicity for reliable software speculation

 Naveen Neelakantam, Ravi Rajwar, Suresh Srinivas, Uma Srinivasan, Craig Stachowiak

June 2007 **ISCA '07: Proceedings of the 34th annual international symposium**
on computer architecture

Publisher: ACM

Full text available:  Pdf (805.55 KB)

Additional Information: [full citation](#), [abs](#)

Bibliometrics: Downloads (6 Weeks): 13, Downloads (12 Months): 121, Downl

Speculative compiler optimizations are effective in improving both single-threaded performance and system-wide energy consumption, but their implementation introduces significant complexity in the optimization scope, and ...

Keywords: Java, atomicity, checkpoint, isolation, optimization, speculative

Also published in:

June 2007 **SIGARCH Computer Architecture News** Volume 35 Issue 2

15 Design and evaluation of dynamic optimizations for a Java just-in-time compiler

 Toshio Suganuma, Toshiaki Yasue, Motohiro Kawahito, Hideaki Komatsu, Toshiyuki Matsunaga

July 2005 **Transactions on Programming Languages and Systems**

Publisher: ACM  [Request Permissions](#)

Full text available:  Pdf (1.60 MB)

Additional Information: [full citation](#), [abs](#)

Bibliometrics: Downloads (6 Weeks): 21, Downloads (12 Months): 219, Downl

The high performance implementation of Java Virtual Machines (JVM) aims toward employing a dynamic compilation system on the basis of online

between the compilation overhead ...

Keywords: JIT compiler, Recompilation, adaptive optimization, code space directed method inlining

16 Revisiting Out-of-SSA Translation for Correctness, Code Quality and Performance
 Benoit Boissinot, Alain Darte, Fabrice Rastello, Benoit Dupont de Dinechin,
 March 2009 **CGO '09: Proceedings of the 2009 International Symposium on Code Generation and Optimization**

Publisher: IEEE Computer Society

Full text available:  Pdf (287.20 KB)

Additional Information: [full citation](#), [abstract](#)

Bibliometrics: Downloads (6 Weeks): 19, Downloads (12 Months): 80, Downloaded by 1 user

Static single assignment (SSA) form is an intermediate program representation that can be performed with fast and easy-to-implement algorithms. However, in certain situations where the SSA variables ...

Keywords: SSA form, Compilers, JIT-compilation

17 CodeBricks: code fragments as building blocks

 Giuseppe Attardi, Antonio Cisternino, Andrew Kennedy

June 2003 **PEPM '03: Proceedings of the 2003 ACM SIGPLAN workshop on Program manipulation**

Publisher: ACM 

Full text available:  Pdf (294.34 KB)

Additional Information: [full citation](#), [abstract](#)

Bibliometrics: Downloads (6 Weeks): 6, Downloads (12 Months): 38, Downloaded by 1 user

We present a framework for code generation that allows programs to be composed at a high level while the joining and splicing of executable code is carried out automatically. The framework introduces ...

Keywords: domain specific language, generative programming, metaprogramming, program generation, program transformation, reflection

Also published in:

October 2003 **SIGPLAN Notices** Volume 38 Issue 10

18 Techniques for obtaining high performance in Java programs

 Iffat H. Kazi, Howard H. Chen, Berdenia Stanley, David J. Lilja

September 2000 **Computing Surveys (CSUR)**, Volume 32 Issue 3

Publisher: ACM 

Full text available:  Pdf (816.13 KB)

Additional Information: [full citation](#), [abstract](#)

Bibliometrics: Downloads (6 Weeks): 77, Downloads (12 Months): 487, Downloaded by 1 user

This survey describes research directions in techniques to improve the performance of Java programming language. The standard technique for Java execution is interpreted, which limits the portability of programs. ...

Keywords: Java, Java virtual machine, bytecode-to-source translators, interpreters, just-in-time compilers

19 SciSim: a software performance estimation framework using source

 Zhonglei Wang, Antonio Sanchez, Andreas Herkersdorf

June 2008 **WOSP '08**: Proceedings of the 7th international workshop on

Publisher: ACM 

Full text available:  Pdf (1.18 MB)

Additional Information: [full citation](#), [abs](#)

Bibliometrics: Downloads (6 Weeks): 17, Downloads (12 Months): 93, Download

Recently, software performance estimation based on source code instrumentation literature. It achieves significant speedup without compromising accuracy. However, much work still ...

Keywords: debugging information, microarchitecture, software performance instrumentation

20 Demystifying magic: high-level low-level programming

 Daniel Frampton, Stephen M. Blackburn, Perry Cheng, Robin J. Garner, David R. O'Hallaron

March 2009 **VEE '09**: Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments

Publisher: ACM 

Full text available:  Pdf (479.11 KB)

Additional Information: [full citation](#), [abs](#)

Bibliometrics: Downloads (6 Weeks): 35, Downloads (12 Months): 196, Download

The power of high-level languages lies in their abstraction over hardware, security, better reliability, and lower development costs. However, opaque systems programmers, ...

Keywords: debugging, intrinsics, jikes rvm, magic, mmtk, systems programming

The ACM Portal is published by the Association for Computing Machinery. Copyright ©

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)